KLASIFIKASI *POLYCYSTIC OVARY SYNDROME* BERDASARKAN CITRA ULTRASONOGRAFI MENGGUNAKAN *PRINCIPAL COMPONENT ANALYSIS* DAN *NAÏVE BAYES* UNTUK MEMBANTU MENDETEKSI KESUBURAN WANITA

POLYCYSTIC OVARY SYNDROME CLASSIFICATION BASED ON ULTRASONOGRAPHY IMAGE USING PRINCIPAL COMPONENT ANALYSIS AND NAÏVE BAYES TO HELP WOMAN FERTILITY DETECTION

Nanda Budi Prayuga[1], Mohamad Syahrul Mubarok, S.T., M.Sc.[2], Prof. Dr. Adiwijaya, S.si., M.Si.[3]
[1,2,3]Prodi S1 Teknik Informatika, Fakultas Informatika, Universitas Telkom
[1]nbprayuga@gmail.com, [2]msyahrulmubarok@telkomuniversity.ac.id, [3]adiwijaya@telkomuniversity.ac.id

Abstrak

Polycystic Ovary Syndrome (PCOS) adalah kelainan sindrom yang diderita wanita di sistem reproduksinya, seseorang dikatakan menderita *Polycystic Ovary Syndrome* (PCOS) jika ada lebih dari 12 *follicle* berukuran 2-9 mm atau bertambah besarnya volume *follicle* di ovarium hingga lebih dari 10 cm3[3]. Saat ini untuk mendeteksi *Polycystic Ovary Syndrome* (PCOS) dokter harus melakukan scan USG, dan secara manual menghitung jumlah *follicle* yang ditandai dengan area hitam di gambar. Pada penelitian sebelumnya [1, 3, 5] hanya berfokus pada peningkatan kualitas citra dan juga pendeteksian ukuran dan jumlah *follicle* untuk mempermudah tenaga medis melihat *follicle* dan menentukan diagnosis pasien. Sehingga saat ini dokter membutuhkan suatu sistem yang dapat membantunya dalam mendiagnosis *Polycystic Ovary Syndrome* (PCOS) secara otomatis berdasarkan citra USG untuk pendeteksian kesuburan wanita. Pada tugas akhir ini dibangun sebuah sistem klasifikasi dengan menggunakan kombinasi metode *Principal Component Analysis* (PCA) yang berfungsi sebagai dimensi reduksi dan *Naïve Bayes* yang merupakan salah satu turunan dari *Bayesian Network* sebagai *classifier*nya. Dari hasil pengujian menggunakan metode *k-fold cross validation* dengan k=8 dan pengujian dilakukan sebanyak 50x pengujian, dapat dilihat sistem yang dibangun dengan menggunakan metode *Principal Component Analysis* (PCA) dan *Naïve Bayes*, memiliki performansi rata-rata *F1 Score* tertinggi sebesar 84.76%, dengan parameter uji jumlah distribusi data ditiap kelas pada data *training* masing-masing 40 gambar, dan jumlah *principal component* sebanyak 53 serta data telah dinormalisasi.

Kata Kunci: Polycystic Ovary Syndrome, ovarium, citra USG, follicle, Naïve Bayes, Principal Component Analysis, Cross Validation, Imbalanced Data, Normalisasi.

Abstract

Polycystic Ovary Syndrome (PCOS) is a disorder syndrome suffered by women in the reproductive system, a woman diagnosed to have Polycystic Ovary Syndrome (PCOS) if there are more than 12 follicles of 2-9 mm or increase in volume of follicles in the ovaries up to more than 10 cm3[3]. Nowadays to detect Polycystic Ovary Syndrome (PCOS), the doctor should perform an ultrasound scan, and manually count the number of follicles marked with the black area in the image, the previous paper [1, 3, 5] only focused on improving image quality and the detection of the size and number of follicles to facilitate medical personnel to see the follicle and determine the patient's diagnosis. Otherwise doctors need a system that can help in diagnosing Polycystic Ovary Syndrome (PCOS) automatically based on ultrasound images for female fertility detection.In this paper, we propose to build a classification system that help in diagnosing Polycystic Ovary Syndrome (PCOS) automatically based on ultrasound image, using a combination of Principal Component Analysis (PCA) method that serves as reduction dimension and Naïve Bayes which is one derivative of Bayesian Network as its classifier. From the test result using k-fold cross validation method with k=8 and 50x testing, showed that the system we build using Principal Component Analysis (PCA) and Naïve Bayes method was succesfully implemented with highest performance of Average F1 Score is 84.76%, with testing parameter: the data amount in each class on the training dataset is 40, and the number of principal component is 53 and the is normalized.

Keywords: Polycystic Ovary Syndrome, ovarium, citra USG, follicle, Naïve Bayes, Principal Component Analysis, Cross Validation, Imbalanced Data, Normalization.

---

1. Pendahuluan

Polycystic Ovary Syndrome (PCOS) adalah kelainan sindrom yang diderita wanita di sistem reproduksinya, seseorang dikatakan menderita PCOS jika ada lebih dari 12 *follicle* berukuran 2-9 mm atau bertambah besarnya volume *follicle* di ovarium hingga lebih dari 10 cm3[2]. Menurut data dari *National Institutes of Health* (NIH) lebih dari 5 juta di USA menderita PCOS. Seseorang yang menderita PCOS akan mengalami masalah pada kesuburannya serta mempunyai hormon androgen dan insulin yang tinggi. Efeknya orang tersebut akan beresiko menderita diabetes tipe 2, kolestrol tinggi, dan tekanan darah tinggi. Sehingga dari data departemen kesehatan Amerika Serikat dibutuhkan lebih dari $4 juta dolar per tahun untuk menganangi permasalahan ini[2].

Saat ini untuk mendeteksi PCOS dokter harus melakukan scan USG dan secara manual menghitung jumlah *follicle* yang di tandai dengan area hitam di gambar. Tentu saja hal ini membutuhkan ketelitian dan pengamatan yang jeli selain ukuran *follicle* yang kecil, *follicle* biasanya tersamarkan dengan obyek lainnya seperti usus atau pembuluh darah. Pada penelitian sebelumnya [1, 3, 5] berfokus pada peningkatan kualitas citra dan juga pendeteksian ukuran dan jumlah *follicle* untuk mempermudah tenaga medis melihat *follicle* dan menentukan diagnosis pasien.

Kombinasi penggunaan *Principal Component Analysis* (PCA) dan *Naïve Bayes* sangat berpotensi digunakan dalam pemecahan masalah deteksi PCOS. *Principal Component Analysis* (PCA) dipilih karena dapat mereduksi dimensi dari data set dengan sesedikit mungkin menghilangkan informasi yang ada didalamnya, sedangkan *Naïve Bayes* adalah salah satu metode *learning* yang menggunakan model probabilistik dan aturan *Bayes* untuk proses inferensinya[14]. Metode ini dipilih karena keefektifannya dalam melakukan klasifikasi, walaupun dalam teorinya di asumsikan tiap fitur independen (*naïve*). Hasil akhirnya diharapkan dengan menggunakan metode *Naïve Bayes* didapatkan sistem yang powerful, efisien, dan memiliki tingkat performansi yang tinggi untuk membantu dokter dalam mendiagnosis *Polycystic Ovary Syndrome* (PCOS) untuk pendeteksian kesuburan wanita.

2.1. *Ovarium*

Untuk mengetahui ukuran ovarium normal dapat dilihat saat masa menstruasi, melalui sebuah tes telah dilakukan di Creighton University School Of Medicine didapat hasil dimana ciri-ciri ovarium normal yaitu : ukuran awal *follicle* 2-4 mm, lalu akan terus tumbuh mencapai 10 mm dihari 8-9 dan mencapai ukuran 18-24mm pada hari ke 14[4]. Sedangkan salah satu kriteria seseorang menderita *Polycystic Ovary Syndrome* (PCOS) adalah ada lebih dari 12 *follicle* berukuran 2-9 mm atau bertambah besarnya volume *follicle* di ovarium hingga lebih dari 10 cm3[2]. Ovarium normal memiliki volume kurang dari 10 cm3 (7.94 ± 2.34 cm3), sedangkan ovarium penderita *Polycystic Ovary Syndrome* (PCOS) memiliki volume melebihi 10 cm3 (14.04 ± 7.36 cm3) [8, 9].

2.2. *Naïve Bayes*

Naïve Bayes adalah salah satu metode *learning* yang menggunakan model probabilistik dan aturan Bayes untuk proses inferensinya[14]. Ciri utama dari algoritma *Naïve Bayes* yaitu adanya asumsi bahwa tiap fitur yang ada di data adalah independen. Persamaan yang digunakan pada model Naïve Bayes bisa dilihat pada persamaan (1) di bawah ini:

$$p(Cn|x) = p(C)\prod_{1<i<k} p(x_i|Cn) \qquad (1)$$

Karena data yang akan diolah adalah data kontinyu, maka formula diatas bisa dijabarkan di persamaan (2) berikut ini:

$$p(Cn|x) = p(C)\prod_{1<i<k} \frac{1}{\sqrt{2\pi\sigma_C^2}} e^{-\frac{(x_i-\mu_C)^2}{2\sigma_C^2}} \qquad (2)$$

$p(Cn|x)$ adalah *posterior probability* data terhadap kelas, $p(C)$ adalah *prior probability* suatu kelas dan $p(x_i|Cn) = \frac{1}{\sqrt{2\pi\sigma_C^2}} e^{-\frac{(x_i-\mu_C)^2}{2\sigma_C^2}}$ adalah *likelihood probability* sebuah fitur terhadap kelas C.

2.3. *Imbalance* Data

Imbalance Data adalah permasalah dalam *machine learning* dimana total data antar kelas tidak sama besarnya[10]. Data set yang jumlahnya lebih besar dibandingkan jumlah data set di kelas lainnya disebut *major*

class, sedangkan data set kelas yang jumlahnya paling sedikit disebut *minor class*[10]. Permasalahan yang biasanya dihadapi saat membangun sistem klasifikasi dengan data yang tidak seimbang adalah kesalahan klasifikasi terhadap *minor class*.Untuk itu dibutuhkan metode untuk melakukan balancing data. Menurut Haibo He, dan Edwardo A. Garcia [7] beberapa solusi yang didapat dari permasalah tersebut yaitu : *Random Oversampling, Random Undersampling, Synthetic Sampling with Data Generation* (SMOTE), *Adaptive Synthetic Sampling* (ADAYS).

2.4. *Principal Component Analysis (PCA)*

Menurut J.Shlens [9], *Principal Component Analysis (PCA)* adalah metode *non-parametric* yang digunakan untuk mengekstrak informasi yang revelan dari dataset yang besar. *Principal Component Analysis (PCA)* juga dapat digunakan untuk mereduksi data yang kompleks ke data yang lebih kecil dimensinya.

2.5. *K-Fold Cross Validation*

K-Fold Cross Validation adalah metode statistik untuk mengevaluasi dan membandingkan algoritma pembelajaran dengan membagi data menjadi dua bagian, yaitu data *training* dan data *testing* [11]. Seluruh data akan diacak dan dibagi menjadi K buah subset dengan ukuran yang sama. Performansi dari sistem yang dibangun akan dihitung dari rata-rata hasil performansi dari K *iterasi*.

2.6. *Normalization*

Normalization atau normalisasi adalah proses transformasi dimana sebuah atribut numerik diskalakan dalam range tertentu, seperti -1.0 sampai 1.0, atau 0.0 sampai 255.0 [8]. Beberapa metode yang di gunakan dalam normalisasi data yaitu: *Min-Max Normalization* dan *Z-Score Normalization.*

2.7. *Perhitungan Performansi*

Untuk mengetahui performansi dari sistem yang sudah dibuat maka perlu dihitung *recall*, *precision*, dan *f1 measure*. Untuk melakukan perhitungan ini di butuhkan *confusion matrix*, seperti dapat dilihat di Tabel 2.1:

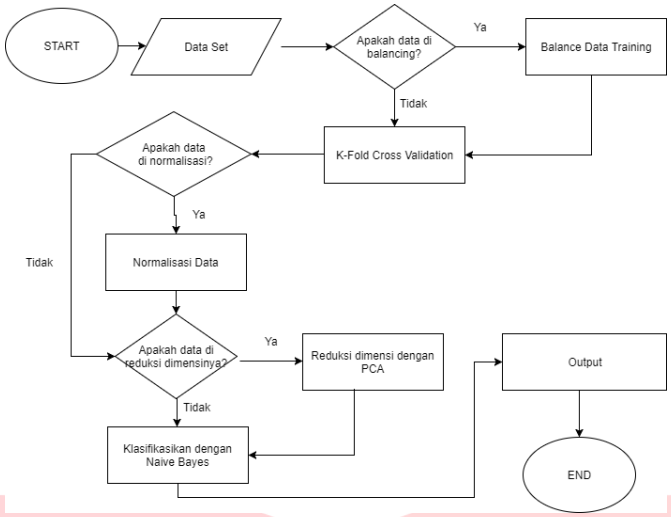
Tabel 2. 1 Confusion Matrix

	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

Sedangkan rumus perhitungannya sendiri yaitu :

$$Recall = \frac{TP}{TP+FN} \quad (3) \qquad Precision = \frac{TP}{TP+FP} \quad (4) \qquad F1\ Score = \frac{2PR}{P+R} \quad (5)$$

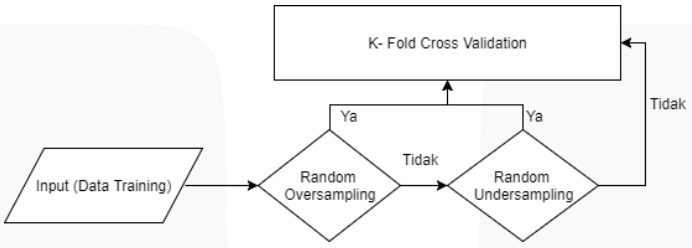
3. Gambaran Umum Sistem



Gambar 3. 1 Gambaran Umum Sistem

Seperti yang bisa dilihat di Gambar 3.1, sistem yang akan dibangun dibagi menjadi lima bagian utama, yaitu: *balancing* data, *K-Fold Cross Validation*, reduksi dimensi, normalisasi data dan klasifikasi. Sebelum melalui kelima proses tersebut data set dibagi menjadi dua bagian, yaitu: data positif *Polycystic Ovary Syndrome* (PCOS) dan data negatif *Polycystic Ovary Syndrome* (PCOS), setiap data set yang berukuran 150x100 akan diubah menjadi vektor berukuran 1x15.000 yang nantinya akan dijadikan inputan dari sistem yang dibangun.

3.1. Perancangan *Balancing* Data



Gambar 3. 2. Alur Diagram Balancing Data

Sebelum dilakukan proses reduksi dimensi dengan PCA, input data set akan di*balancing* dengan salah satu dari dua metode di tersebut (Gambar 3.2). Menurut Haibo He, dan Edwardo A. Garcia [7], penjelasan dari masing-masing metode di atas (Gambar 3.2) adalah sebagai berikut:

a) Random Oversampling

Random Oversampling adalah suatu mekanisme untuk menambah data set *minor class* dengan cara:

1. Diasumsikan data *mayor class* berjumlah 40 gambar, lalu pilih data yang tersedia dari *minor class* secara acak.
2. Salin data yang sudah di pilih dan tambahkan ke data set *minor class* sampai jumlah distribusi data *minor* class 40 gambar.

b) Random Undersampling

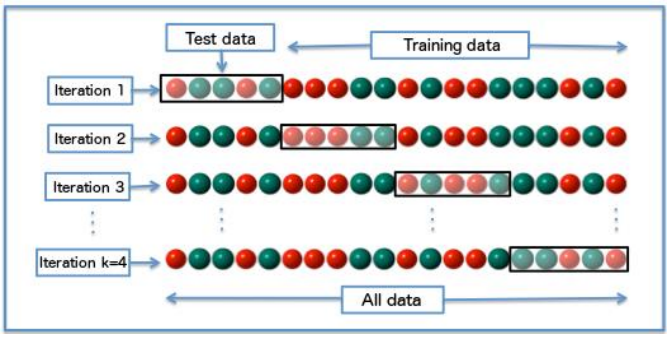
Kebalikan dari *random oversampling*, *random undersampling* adalah suatu mekanisme untuk mengurangi data set *mayor class* dengan cara:

1. Diasumsikan data *minor class* berjumlah 14 gambar, lalu pilih data yang tersedia dari *mayor class* secara acak
2. Hilangkan data yang sudah di pilih dari data set *mayor class* sampai jumlah distribusi data di *mayor class* 14 gambar.

3.2. Perancangan *K-Fold Cross Validation*

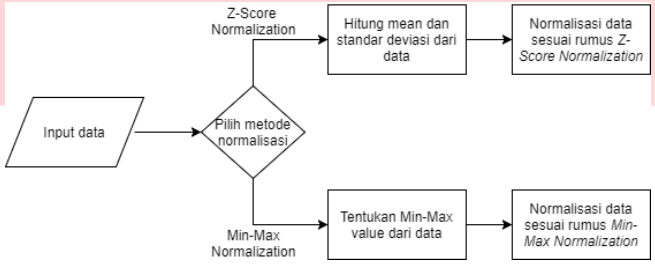
Diasumsikan K=8, maka seluruh data akan diacak dan dibagi menjadi 8 buah subset dengan ukuran yang sama. Setiap subset akan menjadi data *testing* sebanyak satu kali dan akan menjadi data *training* sebanyak K-1

kali. Performansi dari sistem yang dibangun akan dihitung dari rata-rata hasil performansi dari 8 iterasi.Untuk lebih jelasnya dapat dilihat di Gambar 3.3.



Gambar 3. 3 K-Fold Cross Validation (Sumber Wikipedia.org[15])

3.3. Perancangan Normalisasi Data



Gambar 3.4. Alur Diagram Normalisasi Data

Penjelasan dari masing-masing proses *Z-Score Normalization* dijelaskan dibawah ini :

1. *Load* data hasil dari pemrosesan sebelumnya.
2. Ubah setiap data set yang berukuran 150x100 menjadi vektor berukuran 1x15.000.
3. Sehingga nanti akan kita dapat matriks berukuran 80x15.000 yang kita namakan matriks Z (jumlah data 80 gambar).
4. Hitung *mean* dan *standar deviasi* dari data.
5. Langkah kedua yaitu menormalisasi sesuai persamaan (6) di bawah ini:

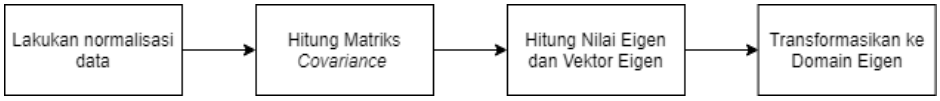
$$x' = \frac{x - min_x}{max_x - min_x}\left(new_{max_x} - newmin_x\right) + newmin_x \quad (6)$$

Penjelasan dari persamaan (6) di atas :

x' = nilai baru dari *x*
min_x = minimum value dari *x*
max_x = maksimum value dari *x*
new_{max_x} = maksimum value baru dari *x*
$newmin_x$ = maksimum value baru dari *x*

3.4. Perancangan *Principal Component Analysisi (PCA)*

Perancangan sistem *Principal Component Analysis (PCA)* dalam sistem yang akan dibangun dapat dilihat dialur diagram pada Gambar 3.5.



Gambar 3.5. Alur Diagram PCA

Penjelasan dari masing-masing proses pada diagram PCA di atas (Gambar 3.5) adalah sebagai berikut.

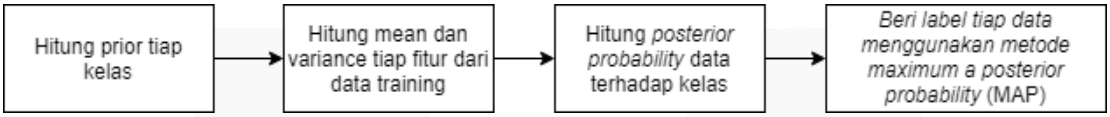
a) Lakukan Normalisasi Data
Tahap pertama yang harus dilakukan dalam proses PCA yaitu melakukan normalisasi data dengan cara:
1. *Load* data hasil dari pemrosesan sebelumnya berupa vektor dari matriks 80x15.000 (diasumsikan jumlah data 80 gambar).
2. Hitung *mean* dari tiap fitur.
3. Kurangi setiap fitur dengan *mean* diperhitungan sebelumnya.

b) Hitung Matriks *Covariance*
Dalam proses ini perhitungan matriks *covariance* dilakukan dengan cara:
1. Kita namakan matriks 80x15.000 adalah matriks Z
2. Kalikan transpose matriks Z' (15.000x80) dengan matriks Z 80x15.000 hasilnya adalah matriks berukuran 15.000x15.000

c) Hitung Nilai Eigen dan Vektor Eigen
Hasil dari perhitungan matriks *covariance* dilangkah sebelumnya kemudian diolah lagi untuk mendapatkan nilai eigen dengan ukuran 15.000x1 dan vektor eigen nya berukuran 15.000x15.000.

d) Tranformasikan ke Domain Eigen
Setelah nilai eigen dan vektor eigen didapat, lalu dilakukan transformasi ke domain eigen dengan cara:
1. Urutkan vektor eigen berdasarkan nilai eigen terbesar.
2. Lalu ambil *x principal component* dari vektor eigen tersebut (nilai *x* yang di spesifikasikan yaitu 53), sehingga didapat vektor eigen berukuran 15.000x53.
3. Langkah terakhir transformasikan matriks original ke domain eigen dengan cara mengalikan transpose dari dataset (matriks Z') dengan transpose vektor eigen (matriks E') dari perhitungan sebelumnya. Hasil akhir yang didapat yaitu dimensi dari data set yang sudah tereduksi dan *feature vector* berukuran 80x53.

3.5. Perancangan *Naïve Bayes*



Gambar 3.6. Alur Diagram Naïve Bayes

Pada pengerjaan klasifikasi menggunakan *Naïve Bayes* ditugas akhir ini, diasumsikan bahwa kolom pada matriks merupakan fitur dari data set. Penjelasan dari masing-masing proses pada diagram *Naïve Bayes* (Gambar 3.6) adalah sebagai berikut.

1. Hitung *Prior* tiap Kelas
Tahap pertama yang harus dilakukan adalah menghitung *prior* dari tiap kelas, diasumsikan data positif PCOS berjumlah 40 gambar dan negatif PCOS berjumlah 40 gambar, sehingga total data set berjumlah 80 gambar. Cari prior per kelas dengan rumus:

$$\mathrm{p(C)} = \frac{N_c}{N} \quad (7)$$

Maka di dapat *prior* tiap kelas, yaitu:
Prior kelas positif PCOS: $\mathrm{p}(+) = \frac{40}{80} = 0.5$
Prior kelas negatif PCOS: $\mathrm{p}(-) = \frac{40}{80} = 0.5$

2. Hitung *Mean* dan *Variance* tiap Fitur dari Data Training
Dalam proses ini akan dicari *mean* dan *variance* tiap fitur data training yang nantinya akan digunakan sebagai parameter *learning* dengan cara:
   a. *Load* data *training* dari proses sebelumnya, berupa matriks berukuran 80x53.
   b. Pisahkan data positif PCOS dan data negatif PCOS dari data training, sehingga akan ada matriks dari data positif PCOS berukuran 40x53 dan matriks dari data negatif PCOS berukuran 40x53.
   c. Cari *mean* dan varian tiap fitur dari kedua kelas data, sehingga setiap kelas akan memiliki *mean* dan varian berukuran 53x1.

3. Hitung *Posterior Probability* Data terhadap Kelas
Proses perhitungan *posterior probability* data terhadap kelas yang dilakukan sudah dijabarkan seperti berikut:
   a. *Load* data *testing* yang sudah melewati proses PCA berukuran 1x53
   b. Karena data berbentuk kontinyu, maka untuk mencari *likelihood* dari tiap kelas dapat dilakukan dengan cara mengalikan setiap fitur yang yang ada di data *testing*, sesuai rumus di bawah ini:

$$p(x_i|Cn) = \frac{1}{\sqrt{2\pi\sigma_C^2}} e^{-\frac{(x_i-\mu_C)^2}{2\sigma_C^2}} \quad (9)$$

x_i adalah fitur/kolom ke *i* dari data *testing*, sedangkan σ_C^2 adalah varian ke *i* dari tiap kelas, dan μ_c adalah *mean* ke *i* dari tiap kelas. Setelah didapat *likelihood* dari setiap kelas, selanjutnya dihitung *posterior probability* dari setiap kelas dengan mengalikan *prior* dan *likelihood* nya. Hasil akhirnya akan didapat *posterior* untuk kelas positif PCOS dan *posterior* untuk kelas negatif PCOS.

4. Beri Label tiap Data menggunakan *Metode Maxium a Posterior Probability (MAP)*
   Setelah nilai *posterior probability* data terhadap kelas didapat, lalu dilakukan *labeling* data dengan dengan menggunakan metode *maximum a posterior probability* (MAP) seperti yang sudah di jelaskan di bab 2. Nilai tertinggi dari hasil perhitungan *posterior* akan menentukan label dari data tersebut.

4. Pengujian dan Analisis

Sistem yang dibangun adalah untuk mengetahui performansi dari sistem berdasarkan skenario pengujian yang sudah dibuat menggunakan data set yang sudah disediakan. Sistem yang dibuat diharapkan memiliki performansi yang bagus dengan tingkat klasifikasi yang tinggi sehingga membantu dokter dalam mendiagnosis *Polycystic Ovary Syndrome* (PCOS) untuk pendeteksian kesuburan wanita.

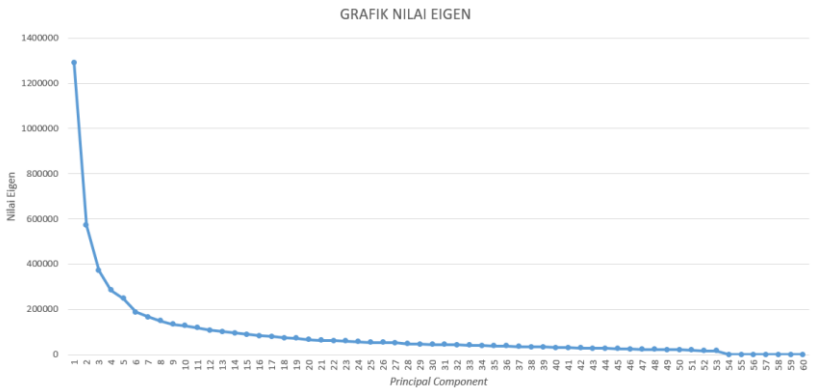
4.1. Pengujian Skenario I

Tabel 4. 1 Observasi Jumlah Distrubusi Data Setiap Kelas

	Jumlah data 40-40	Jumlah data 14-40	Jumlah data 14-14
F1 Score	84.50%	38.75%	28.16%
Recall	76.15%	49.23%	23.59%
Precision	99.00%	38.24%	46.38%

Berdasarkan grafik pada Tabel 4.1 dapat dilihat bahwa jumlah data tiap kelas mempengaruhi performansi sistem yang dibuat, hal ini dapat dilihat dari nilai *recall*, *precision*, dan *f1 score* yang berbeda. Menggunakan metode *Random Oversampling* untuk memperbanyak data dan *K-Fold Cross Validation* dengan k=8 serta dilakukan percobaan sebanyak 50 kali, parameter uji dengan jumlah data kelas negatif PCOS 40 dan jumlah data kelas positif PCOS 40 memberikan hasil terbaik dengan nilai *recall* sebesar 76.15%, *precision* 99.00% dan *f1 score* 84.50%. Jumlah distribusi data di setiap kelas mempengaruhi performansi karena metode klasifikasi *Naïve Bayes* adalah metode klasifikasi yang sensitif dengan jumlah data set tiap kelasnya yang harus seimbang, dengan jumlah data set yang seimbang di setiap kelasnya menjadikan nilai *prior* antar kelas akan sama besarnya $p(C) = 0.5$. Selain itu dengan jumlah data *training* yang semakin banyak, maka sistem yang dibangun bisa melakukan proses *learning* terhadap data yang lebih beragam. Hal tersebut juga mempengaruhi nilai *mean* dan varian yang nantinya akan digunakan sebagai parameter *learning* pada tahap klasifikasi.

4.2. Pengujian Skenario II



Gambar 4. 1. Grafik Nilai Eigen

Tabel 4. 2 Nilai Eigen 1-60

Principal Component	Nilai Eigen									
	1	2	3	4	5	6	7	8	9	10
	1292486	571133.3	371341.1	285117.9	248015.5	188456.2	166803.8	148993.4	134404.7	127934.7
10	117583.6	108042.6	101069.4	94869.78	88263.39	82613.72	79780.61	73653.06	71713.38	65640.81
20	62294.51	60983.49	59898.92	56964.99	53569.14	52900.55	52075.3	47981.86	46551.1	45021.79
30	44047.68	43139.1	41391.7	39264.25	38659.77	37485.41	35557.12	34089.95	33440.56	30962.52
40	30558.55	28655.88	27216.87	26545.72	26020.15	24135.28	22944.26	22304.59	21086.09	20469.13
50	20083.73	17128.67	15749.55	0.190546	0.142884	0.01706	0.010676	0.008359	0.007935	0.005308

Tabel 4. 3 Observasi Principal Component 10-200

	10	20	30	40	50	60	70	80	90	100
F1 Score	77.77%	67.10%	64.63%	65.53%	86.66%	87.02%	87.02%	87.02%	87.02%	87.02%
Recall	74.39%	86.45%	93.90%	92.94%	79.45%	79.25%	79.25%	79.25%	79.25%	79.25%
Precision	86.31%	57.52%	51.29%	53.02%	98.66%	99.50%	99.50%	99.50%	99.50%	99.50%

	110	120	130	140	150	160	170	180	190	200
F1 Score	87.02%	87.02%	87.02%	87.02%	87.02%	87.02%	87.02%	87.02%	87.02%	87.02%
Recall	79.25%	79.25%	79.25%	79.25%	79.25%	79.25%	79.25%	79.25%	79.25%	79.25%
Precision	99.50%	99.50%	99.50%	99.50%	99.50%	99.50%	99.50%	99.50%	99.50%	99.50%

Tabel 4. 4 Observasi Principal Component 45-65

	45	46	47	48	49	50	51	52	53	54	55
F1 Score	65.84%	68.76%	73.99%	79.23%	82.93%	84.42%	84.38%	84.59%	84.76%	84.74%	84.56%
Recall	87.89%	85.56%	82.56%	79.87%	77.47%	76.82%	76.24%	76.28%	76.45%	76.41%	76.19%
Precision	55.50%	61.89%	73.14%	85.12%	94.36%	97.87%	98.48%	98.97%	99.00%	99.00%	99.00%

	56	57	58	59	60	61	62	63	64	65	
F1 Score	84.46%	84.55%	84.51%	84.52%	84.50%	84.53%	84.55%	84.46%	84.44%	84.45%	
Recall	76.05%	76.20%	76.12%	76.17%	76.15%	76.22%	76.20%	76.10%	76.08%	76.09%	
Precision	99.00%	99.00%	99.00%	99.00%	99.00%	99.00%	99.00%	99.00%	99.00%	99.00%	

Didalam teori PCA dikatakan bahwa semakian besar nilai eigen suatu komponen maka semakin besar juga informasi yang terkandung didalamnya. Ada beberapa cara yang bisa dilakukan untuk mencari jumlah *principal components* paling optimal diantaranya yaitu: *Scree test* [12], dan *cross validation* [13]. Menurut Cattell [12] *scree test* adalah teknik grafis yang digunakan untuk menggambarkan nilai eigen dari komponen PCA sehingga dari grafis yang ditampilkan dapat diketahui jumlah *principal component* paling optimal. Seperti yang terlihat di Tabel 4.2. Dari observasi yang dilakukan dengan menggunakan *scree test* dapat dilihat bahwa jumlah *principal component* paling optimal adalah 53 komponen (nilai eigen komponen ke 54 sangat kecil = 0.190546). Setelah diketahui jumlah *principal component* paling optimal menggunakan *scree test*, langkah selanjutnya kita lakukan validasi dengan *cross validation*. Penjelasan proses yang dilakukan saat validasi menggunakan *cross validation* dijabarkan sebagai berikut:

1. Dilakukan observasi dengan nilai *principal component* dari 10, 20, 30, .., 200 dengan percobaan sebanyak 50 kali. Pada Tabel 4.3 dapat dilihat bahwa hasil *F1 Score* naik saat menggunakan *principal component* sebanyak 50, dan nilai *F1 Score* stabil disaat nilai *principal component* 60. Dari hasil percobaan di atas dapat ditarik kesimpulan bahwa penggunaan *principal component* paling optimal ada direntang 50-60 komponen.
2. Setelah dipercobaan sebelumnya diketahui bahwa *principal component* paling optimal ada di rentang 50-60 komponen, maka di percobaan kedua ini akan di lakukan observasi dengan nilai *principal component* dari 45,46,47, .., 65 dengan percobaan sebanyak 50 kali. Pada Gambar Tabel 4.4 dapat dilihat bahwa hasil *F1 Score* terbaik muncul saat menggunakan *principal component* sebanyak 53.

Berdasarkan observasi, penggunaan PCA dan jumlah *principal components* yang digunakan terbukti mempengaruhi performansi sistem yang dibuat, hal ini dapat dilihat dari nilai *recall*, *precision*, dan *f1 score* yang berbeda. Penggunaan PCA akan mereduksi dimensi data yang akan diproses dengan sesedikit mungkin menghilangkan informasi yang terkandung di dalamnya, sehingga sangatlah penting untuk mengetahui jumlah *principal component* paling optimal yang bisa digunakan. Sesuai dengan hasil *Scree test* dan *cross validation*, parameter uji dengan jumlah *principal components* = 53 memberikan hasil terbaik dengan nilai *recall* sebesar 76.45%, *precision* 99.00% dan *f1 score* 84.76%.

4.3. Pengujian Skenario III

Tabel 4. 1 Observasi Normalisasi

	80-80	80-80 Normalisasi	40-40	40-40 Normalisasi	14-40	14-40 Normalisasi	14-14	14-14 Normalisasi
F1 Score	81.28%	83.05%	84.76%	82.90%	38.75%	36.88%	28.16%	19.34%
Recall	79.71%	78.54%	76.45%	74.03%	49.23%	41.72%	23.59%	16.01%
Precision	84.76%	89.80%	99.00%	98.50%	38.24%	40.35%	46.38%	31.63%

Berdasarkan Tabel 4.6 dapat dilihat bahwa proses normalisasi mempengaruhi performansi sistem yang dibuat, hal ini dapat dilihat dari nilai *recall*, *precision*, dan *f1 score* yang berbeda. Parameter uji dengan menggunakan proses normalisasi memberikan hasil terbaik dengan nilai *recall* sebesar 76.45%, *precision* 99.00% dan *f1 score* 84.76%, sedangkan yang tidak menggunakan proses normalisasi memberikan hasil performansi dengan nilai *recall* sebesar 82.90%, *precision* 74.03% dan *f1 score* 98.50%. Penggunaan proses normalisasi mempengaruhi performansi karena proses normalisasi akan *merescaling* fitur didalam data sehingga memiliki sifat-sifat seperti distribusi normal dengan rata-rata = 0 dan standar deviasi = 1. Selain itu dengan melakukan proses normalisasi maka fitur-fitur yang ada didalam data memiliki skala yang sama.

5. Kesimpulan

Berdasarkan observasi dan analisis dari percobaan yang telah di lakukan, maka dapat di tarik kesimpulan sebagai berikut :

1. Jumlah distribusi data di setiap kelas mempengaruhi performansi karena metode klasifikasi *Naïve Bayes* adalah metode klasifikasi yang sensitif dengan jumlah data set tiap kelasnya yang harus seimbang. Selain itu dengan jumlah data *training* yang semakin banyak, maka sistem yang dibangun bisa melakukan proses *learning* terhadap data yang lebih beragam. Hal tersebut juga mempengaruhi nilai *mean* dan varian yang nantinya akan digunakan sebagai parameter learning pada tahap klasifikasi. Dari skenario yang sudah diuji, menggunakan metode *Random Oversampling* untuk memperbanyak data, parameter uji dengan jumlah data kelas negatif PCOS 40 dan jumlah data kelas positif PCOS 40 memberikan hasil terbaik dengan nilai *recall* sebesar 76.15%, *precision* 99.00% dan *f1 score* 84.50%.
2. Penggunaan *Principal Component Analysis* (PCA) dan jumlah principal components yang digunakan terbukti mempengaruhi performansi sistem yang dibuat. Penggunaan PCA akan mereduksi dimensi data yang akan diproses dengan sesedikit mungkin menghilangkan informasi yang terkandung di dalamnya, sehingga sangatlah penting untuk mengetahui jumlah *principal component* paling optimal yang bisa digunakan. Sesuai dengan hasil *Scree test* dan *cross validation*, parameter uji dengan jumlah *principal components* = 53 memberikan hasil terbaik dengan nilai *recall* sebesar 76.45%, *precision* 99.00% dan *f1 score* 84.76%.
3. Proses normalisasi terbukti mempengaruhi performansi sistem yang dibuat. Penggunaan proses normalisasi mempengaruhi performansi karena proses normalisasi akan merescaling fitur didalam data sehingga memiliki sifat-sifat seperti distribusi normal dengan rata-rata = 0 dan standar deviasi = 1. Selain itu dengan melakukan proses normalisasi maka fitur-fitur yang ada didalam data memiliki skala yang sama.

Daftar Pustaka

[1] Adiwijaya, B. Purnama, A. Hasyim, M. D. Septiani, U. N.Wisesty, W. Astuti. 2015. Follicle Detection on the USG Images to Support Determination of Polycystic Ovary Syndrome. In Journal of Physics: Conference Series (Vol. 622, No. 1, p. 012027). IOP Publishing.

[2] Guttmacher, A. E. (2012). Final Report The National Institutes of Health Polycystic Ovary Syndrome . NATIONAL INSTITUTES OF HEALTH Evidence-based Methodology Workshop on, 1-40.

[3] Eni Setiawati, Adiwijaya, Tjokorda Agung, 2015. Particle Swarm Optimization on Follicles Segmentation to Support PCOS Detection. 3rd International Conference on Information and Communication Technology (ICoICT).

[4] Creighton University School of Medicine. (2005). Ultrasound of Uterus and Ovary,1-16. http://www.toledoxray.com/sec/Guides/Ultrasound/Ultrasound%20of%20Uterus%20and%20Ovary.pdf

[5] Järvelä, I. Y., Mason, H. D., Sladkevicius, P., Kelly, S., Ojha, K., Campbell, S., & Nargund, G. (2002). Characterization of normal and polycystic ovaries using three-dimensional power Doppler ultrasonography. Journal of assisted reproduction and genetics, 19(12), 582-590.

[6] Balen, A. H., Laven, J. S., Tan, S. L., & Dewailly, D. (2003). Ultrasound assessment of the polycystic ovary: international consensus definitions. Human reproduction update, 9(6), 505-514.

[7] He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, *21*(9), 1263-1284.

[8] Junaedi, H., Budianto, H., Maryati, I., & Melani, Y. DATA TRANSFORMATION PADA DATA MINING.

[9] Shlens, J. (2014). A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*.

[10] Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004). Special issue on learning from imbalanced data sets. *ACM Sigkdd Explorations Newsletter*, *6*(1), 1-6.

[11] Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. In Encyclopedia of database systems (pp. 532-538). Springer US.

[12] Cattell, R. B. (1966). The scree test for the number of factors. Multivariate behavioral research, 1(2), 245-276.

[13] Cangelosi, R., & Goriely, A. (2007). Component retention in principal component analysis with application to cDNA microarray data. Biology direct, 2(1), 2.

[14] Mubarok, M. S., Adiwijaya, & Aldhi, M. D. (2017, August). Aspect-based sentiment analysis to review products using Naïve Bayes. In *AIP Conference Proceedings* (Vol. 1867, No. 1, p. 020060). AIP Publishing.

[15] Wikipedia.org. (2017). https://en.wikipedia.org/wiki/Cross-validation_(statistics)#/media/File:K-fold_cross_validation_EN.jpg

[16] Adiwijaya, 2014, Aplikasi Matriks dan Ruang Vektor, Yogyakarta: Graha Ilmu

[17] Adiwijaya, 2016, Matematika Diskrit dan Aplikasinya, Bandung: Alfabeta